

Edge Based Probabilistic Relaxation for Sub-Pixel Contour Extraction

Toshiro Kubota^{*1}, Terry Huntsberger², and Jeffrey T. Martin³

¹ Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29205 USA
`kubota@cse.sc.edu`

² Jet Propulsion Laboratory, Pasadena, CA 91109
`terry@helios.jpl.nasa.gov`

³ Compaq, Cupertino, CA 95014-2599
`j.martin@compaq.com`

Abstract. The paper describes a robust edge and contour extraction technique under two types of degradation: random noise and aliasing. The technique employs unambiguous probabilistic relaxation to distinguish features from noise and refine their spatial locations at sub-pixel accuracy. The most important component in the probabilistic relaxation is a compatibility function. The paper suggests a function with which the optimal orientation of edges can be derived analytically, thus allowing an efficient implementation of the relaxation process. A contour extraction algorithm is designed by combining the relaxation process and a perceptual organization technique. Results on both synthetic and natural images are given and are promising.

keywords: *feature extraction, relaxation labelling, segmentation*

1 Introduction

Feature extraction is an essential part of most computer vision problems. Many features such as edges and corners are high frequency components and can be easily obscured by noise. Thus, effective feature extraction processes must incorporate some degree of noise removal capability. Another major obstacle against reliable feature extraction is aliasing due to finite sampling of data. The aliasing obscures the spatial location of features. Researchers are continually working to overcome these problems.

Many feature extraction algorithms proposed in literature often assume that noise has been reduced using some standard smoothing techniques such as Wiener filter, Gaussian smoothing, and non-linear diffusion [1, 10, 22]. A problem with handling noise separately from feature extraction process is that it is difficult to determine the necessary amount of smoothing required to remove noise without removing actual features. Even with an optimal amount of smoothing, some of

^{*} Research partially supported by ONE Grant N00014-97-1-1163

subtle features would be lost. Another disadvantage associated with smoothing is that it further obscures the spatial location of features. The trade-off between signal to noise ratio (SNR) and localization accuracy is well known, and linear filter based techniques such as Gaussian smoothing have a theoretical limit in its performance in terms of the SNR and localization.

A more reliable approach is to distinguish features from noise by a localized pattern analysis. The underlying assumption is that features form non-random patterns while noise does not. Also prior knowledge of feature patterns can improve the spatial localization of the features. Such pattern analysis is believed to be a part of the human vision processing as evidenced by vernier hyper acuity and contour “pop-out” [4].

Our research goal is to derive a reliable feature extraction and localization system based on simple localized pattern analysis. Such a system is not only for practical interest but also for theoretical one as it can bring some insight on segmentation mechanism of the human vision system. In this paper, we employ probabilistic relaxation or relaxation labeling ([25, 9]) to filter out random noise and extract high frequency features and their spatial locations from low resolution images. The technique searches for a near optimal edge configuration in terms of its location and orientation at the sub-pixel resolution (i.e., we wish to resolve high-resolution edge contours).

The paper suggests a general approach for designing an edge based compatibility function that is a core ingredient for the relaxation process. It then provides a particular realization of the function that allows computationally efficient procedure for performing the relaxation and achieving a near-optimal edge configuration. We then develop a contour extraction technique that combines the result of the relaxation and perceptual organization technique. The computation is purely local and intrinsically parallel.

The paper is organized as follows. Section 2 provides a brief description of the probabilistic relaxation followed by a detail description of how to design a compatibility function for recovering edge configuration. Section 3 describes how to perform the relaxation process in a computationally efficient manner. Section 4 provides a contour extraction procedure based on the relaxation and perceptual grouping. Section 5 gives some experimental results of the edge localization and contour extraction processes using both synthetic and natural images. Section 5 provides brief discussion on other related works and some relevant neurological evidences. Finally, Section 6 concludes with summary.

2 Probabilistic Relaxation

The technique explores global consistency through local iterative interactions or “relaxation”. It measures local consistency of an object to its neighbor objects based on a collective sum of a simple pair-wise compatibility measure. The measure captures the prior knowledge of the structural or contextual patterns of interest. At each iteration, the configuration of each object is updated so that it is more consistent to its neighbors. The configuration of an object is represented

by the probability distribution of its labels or states. Through iterative local interaction, the process approaches a more globally consistent configuration. The technique has been studied extensively in both theory and implementation [9, 14, 21, 24], and found many applications in image processing and computer vision. [7, 11]

For any applications of the probabilistic relaxation, a compatibility function needs to be designed. It is often defined as $r_{ij}(\lambda_i, \lambda_j)$, of two objects at i and j having labels λ_i and λ_j , respectively. The function quantifies how likely or how compatible the label λ_i of an object at i is to λ_j of another object at j . We can also associate it with the conditional probability $\Pr(\lambda_i|\lambda_j)$. However, the specification of $r_{ij}(\lambda_i, \lambda_j)$ is less constrained than $\Pr(\lambda_i|\lambda_j)$ as the former is allowed to have negative values and $\sum_{\lambda_i} r_{ij}$ does not have to be 1. When λ_i at site i is compatible (incompatible) with λ_j at site j , the compatibility function should have a large (small or negative) value.

Now a support function is defined based on the compatibility function as

$$S_i(\lambda_i) = \sum_j \sum_{\lambda_j} r_{ij}(\lambda_i, \lambda_j) P_j(\lambda_j), \quad (1)$$

where $P_j(\lambda_j)$ is the probability of having label λ_j at j or the Probability Density Function (PDF) of λ_j . The support function measures how likely that the site i is labelled as λ_i given the configuration of its neighbors. At last, the total support function is defined as

$$S = \sum_i \sum_{\lambda_i} S_i(\lambda_i) P_i(\lambda_i) = \sum_i \sum_{\lambda_i} \sum_j \sum_{\lambda_j} r_{ij}(\lambda_i, \lambda_j) P_i(\lambda_i) P_j(\lambda_j) \quad (2)$$

The total support function measures the global consistency of the particular configuration. The objective of the probabilistic relaxation is to maximize the total support by iteratively updating $P_i \forall i$.

In its general form, probabilistic relaxation is not computationally amiable. Difficulties associated with the technique are the following:

1. It is difficult to formulate the compatibility function as the function is 4 dimensional (i, j, λ_i and λ_j) in general.
2. It is not simple to update P_i as it has to be projected onto $\{p_i(\lambda_k), k = 1..K | p_i(\lambda_k) \in [0, 1], \sum_k p_i(\lambda_k) = 1\}$ where K is the number of possible labels, and evaluation of the support function is often computationally expensive. [17, 20]

The second difficulty listed above can be alleviated by using unambiguous relaxation [9]. With unambiguous labeling, the only label allowed for an object at i to take is the one that maximizes S_i . By denoting the index of the label that maximizes S_i as $M(i)$ (i.e. $M(i) = \arg \max_k S_i(\lambda_k)$), the PDF becomes $P_i(\lambda_k) = 1$ if $k = M(i)$ and $P_i(\lambda_k) = 0$ if $k \neq M(i)$. Then the support function becomes

$$S_i(\lambda_{M(i)}) = \sum_j r_{ij}(\lambda_{M(i)}, \lambda_{M(j)}), \quad (3)$$

and

$$S_i(\lambda_k) = 0, k \neq M(i). \quad (4)$$

The total support function is simplified to

$$S = \sum_i \sum_j r_{ij}(\lambda_{M(i)}, \lambda_{M(j)}). \quad (5)$$

Thus, the unambiguous relaxation alleviates the second difficulty listed above at the expense of flexibility in specifying the PDF. However, we still face a problem of designing the compatibility function. In the next sections, we concentrate on designing the compatibility function for application to edge extraction.

2.1 Compatibility function

As described above, the compatibility function captures the prior knowledge of the patterns of interest; thus is heavily dependent on a problem one wants to solve. Here, our interest is to extract edges and their attributes. This section first describes a general approach to design a compatibility function for edges. The approach is based on two assumptions: invariance to an Euclidean transform of the coordinate system and invariance to the global illumination level. Then the section describes a particular realization of the approach for extraction of edges under noisy condition. The realization results in a computationally efficient procedure for maximizing the support function.

First, we consider that an edge is described by three attributes: the location (x, y) , orientation θ , and strength m . It will be represented by a vector, $\mathbf{e}(x, y)$ whose orientation and strength are $\angle \mathbf{e} = \theta$ and $|\mathbf{e}| = m$, respectively. Our framework allows the location and orientation to be treated as label for the relaxation.

We define a compatibility function for edges as

$$r(\mathbf{e}_i, \mathbf{e}_j) = f(i_x, i_y, \angle \mathbf{e}_i, j_x, j_y, \angle \mathbf{e}_j, |\mathbf{e}_j|) \quad (6)$$

where i_x and i_y are the x and y coordinates of \mathbf{e}_i , respectively, as well as j_x and j_y for \mathbf{e}_j . This is a function of 7 variables. Note that the function is not dependent on $|\mathbf{e}_i|$ as we do not treat the edge strength as label.

To simplify the design process, we assume that the compatibility function is invariant to both translation and rotation of the coordinate system. Then, we design a prototype function with $j_x = j_y = 0$ and $\angle \mathbf{e}_j = 0$. Later, this prototype function is translated to (j_x, j_y) and rotated by $\angle \mathbf{e}_j$ to obtain the general form of $r(\mathbf{e}_i, \mathbf{e}_j)$. Then the prototype function can be expressed as

$$r_{ij}(\mathbf{e}_i, \mathbf{e}_j) = f(i_x, i_y, \angle \mathbf{e}_i, |\mathbf{e}_j|). \quad (7)$$

For convenience, we use a polar coordinate in describing (i_x, i_y) . Thus

$$r_{ij}(\mathbf{e}_i, \mathbf{e}_j) = f(d_i, \alpha_i, \angle \mathbf{e}_i, |\mathbf{e}_j|). \quad (8)$$

where $d_i = \sqrt{i_x^2 + i_y^2}$ and $\alpha_i = \arctan(i_y, i_x)$.

In most image capturing environments, the image irradiance is proportional to the scene radiance, thus the edge strength is also proportional to the radiance [27]. Our goal is to obtain a consistent edge configuration based on the structure of objects, as much as possible, without being influenced by the level of illumination. Thus, we impose a constraint that the order of support should be invariant to the constant scaling of the scene illumination. To be more precise mathematically, the ratio of total supports for two different configurations remains constant with respect to the change in the global illumination level.

It can be shown that the above constraint can be satisfied if the compatibility function is proportional to the edge strength. Thus the prototype compatibility function is

$$r_{ij}(\mathbf{e}_i, \mathbf{e}_j) = |\mathbf{e}_j| f(d_i, \alpha_i, \angle \mathbf{e}_i). \quad (9)$$

Note that this is not the only choice for achieving the invariance to the global illumination level. Obviously, making the compatibility function totally independent of the edge strength is another option. However, we found that the scaling of the compatibility measure by the edge strength is important as strong surface discontinuities can strongly influence the neighboring configuration.

The next design step is to decompose f into a product of two terms: r_{loc} that measures the compatibility of the edge location to \mathbf{e}_j , and r_{or} that measures the compatibility of $\angle \mathbf{e}_i$ to \mathbf{e}_j given d_i and α_i . The functions r_{loc} and r_{or} will be called compatibility factors. With this decomposition, f is written as

$$f(d_i, \alpha_i, \angle \mathbf{e}_j) = r_{loc}(d_i, \alpha_i | \mathbf{e}_j) r_{or}(\angle \mathbf{e}_i | d_i, \alpha_i, \mathbf{e}_j). \quad (10)$$

Note that we used $|$ notation as conditional probability to make the meaning of each factor more clearly. One can make an analogy of the decomposition with the product rule of probability. This decomposition can be applied to any function f and does not impose any new constraints on f . However, the design process becomes more tractable by breaking the compatibility relationship into two factors.

Our formulation of the compatibility factors is described next. The design is heavily based on our intuition and other alternatives are possible.

Compatibility Factor r_{loc} . We arrive at our definition of r_{loc} empirically. Colinearity of the Gestalt rules suggest that α_i is most compatible with \mathbf{e}_j when it is equal to $\angle \mathbf{e}_j$ or $\angle \mathbf{e}_j + \pi$ [12]. The degree of the compatibility decreases as α_i deviates from the values. We use $\sin^2(\alpha_i)$ to quantify the deviation. We also assume that the compatibility decreases as the distance between two sites increases. Using the ideas above as guidelines, we suggest the following function for r_{loc} with distributions that are Gaussian in d_i and exponential in $\sin^2(\alpha_i)$.

$$r_{loc}(d_i, \alpha_i) = e^{-\beta \sin^2(\alpha_i) - d_i^2 / 2\sigma^2} \quad (11)$$

where σ^2 and β are parameters for the Gaussian and exponential distributions, respectively.

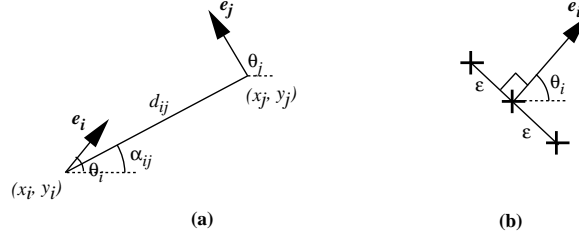


Fig. 1. Geometrical Notations. (a) Notations used in describing the compatibility function. (b) Notations used in describing the support function maximization procedure.

Compatibility Factor: r_{or} . For r_{or} , we use $\cos(\angle e_i - \phi_i(d_i, \alpha_i, e_j))$, where ϕ_i specifies the most compatible $\angle e_i$ to e_j . The function returns 1 when $e_i = \phi_i$ and -1 when $e_i = \phi_i \pm \pi$. It is monotonic between the two extrema.

We found empirically that ϕ_i can be specified further. We collected natural images of various types, measured correlation of gradient angles at different offsets, and obtained PDFs of $\angle \nabla I(x, y) - \angle \nabla I(x + o_x, y + o_y)$ for a distribution of offsets (o_x, o_y) . Note that I represents image data and ∇ is the gradient operator. We found that the PDFs are strongly peaked at 0. Figure 2 shows PDFs of $\angle \nabla I(x, y) - \angle \nabla I(x + o_x, y + o_y)$ at two different offsets: $(o_x, o_y) = (2, 0)$ and $(2, 2)$. The results suggest that the most compatible $\angle e_i$ to e_j is $\angle e_j$. Thus, we set $\phi_i = \angle e_j$.

With the compatibility factors so designed, the prototype compatibility function is

$$r(e_i, e_j) = |e_j| e^{-\beta \sin^2(\alpha_i) - d_i^2/2\sigma^2} \cos(\angle e_i), \quad (12)$$

and for an arbitrary e_j ,

$$r(e_i, e_j) = |e_j| e^{-\beta \sin^2(\alpha_{ij} - \angle e_j) - d_{ij}^2/2\sigma^2} \cos(\angle e_i - \angle e_j), \quad (13)$$

d_{ij} is the distance between (i_x, i_y) and (j_x, j_y) , and α_{ij} is the slope of the line connecting (i_x, i_y) and (j_x, j_y) . See Figure 1(a).

The use of this compatibility factor results in a computationally efficient procedure for maximizing the total support defined in (5). The next section discusses the maximization process.

3 Relaxation Procedure

3.1 Maximizing Support Function

Computational effort is a major consideration for maximizing the total support function. To show this, denote the number of possible edge locations by N_l , and the number of possible edge orientation by N_o . For simplicity, if we allow multiple edges to share the same site, the number of possible configurations for

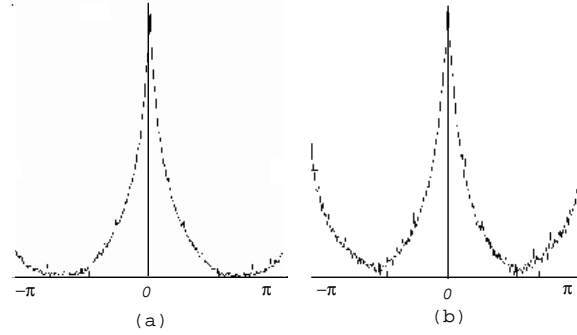


Fig. 2. *Correlation of Edge Orientations.* The plot shows the PDF of $\angle \nabla I(x, y) - \angle \nabla I(x + o_x, y + o_y)$ at different offsets, (o_x, o_y) . Both PDFs show a strong peak at $\angle \nabla I(x, y) - \angle \nabla I(x + o_x, y + o_y) = 0$. (a) $(o_x, o_y) = (2, 0)$ (b) $(o_x, o_y) = (2, 2)$

each edge is $N_l N_o$, which can be quite large for moderate cases. For example, with $N_l = 5 \times 5$ and $N_o = 16$, $N_l N_o = 400$. Then, in order to find $M(i)$ which maximizes S_i in (3), a brute force method requires $N_l N_o$ evaluations of S_i . Thus, the total number of evaluation of the compatibility function is $N_n N_l N_o$ where N_n is the number of neighbor sites contributing to the sum in (3).

Maximizing r_{or} The main reason for using \cos function in r_{or} to interpolate between the two extrema is to reduce the computational burden and, at the same time, increase the resolution of edge orientation. Assume that we are maximizing S_i in terms of $\angle e_i$. Then by using trigonometry identities, $\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b)$ and $A\sin(\theta) + B\cos(\theta) = \sqrt{A^2 + B^2}\cos(\theta + \phi)$, $\phi = \arctan(A/B)$, the support function can be expressed as

$$S_i(\angle e_i) = \sum_j |e_j| r_{loc} \cos(\angle e_i - \angle e_j) = F_i \cos(\angle e_i - \Theta_i) \quad (14)$$

where

$$F_i = \sqrt{(\sum_j |e_j| r_{loc} \cos(e_j))^2 + (\sum_j |e_j| r_{loc} \sin(e_j))^2}$$

and

$$\Theta_i = \arctan \left(\frac{\sum_j |e_j| r_{loc} \sin(e_j)}{\sum_j |e_j| r_{loc} \cos(e_j)} \right).$$

Therefore $\angle e_j = \Theta_i$ maximizes S_i and Θ_i can be computed with N_n evaluations of r_{loc} instead of $N_o N_n$. Also the domain of $\angle e_i$ becomes continuous without any computational penalty.

Another and more visually intuitive interpretation of the above formula may be to consider a vector f_{ij} whose length and angle are $|e_j| r_{loc}$ and $\angle e_j$, respectively, and a vector F_i whose length and angle are F_i and Θ_i , respectively. Then

```

Obtain an initial edge configuration with some gradient operator
do {
  for(each edge element  $i$ ) {
    Compute  $F_i$  at the current location and its neighbor sites.
    Move the edge to the location where  $F_i$  is the largest
    Set the edge orientation to  $\Theta_i$  at the new location
  }
} until convergence

```

Fig. 3. *Procedure for Maximizing the Total Support.*

\mathbf{F}_i can be computed as a vector sum of \mathbf{f}_{ij} . Thus,

$$\mathbf{F}_i = \sum_j \mathbf{f}_{ij} \quad (15)$$

and the $\angle e_j$ that maximizes S_i is $\angle \mathbf{F}_i$.

3.2 Procedure

We propose a local and iterative procedure for obtaining the edge configuration that maximizes S . Because of its local nature, the procedure is not guaranteed to find the global maximum. However, it is computationally efficient, intrinsically parallel and effective in finding a near-optimum solution.

The procedure updates each edge element sequentially. For each e_i , F_i is computed at three different locations: the current edge location and two locations that are ϵ apart from e_i in the direction perpendicular to $\angle e_i$. (See Figure 1(b).) The main reasons for this constrained search are the following. If the estimated edge orientation is correct, the shortest path for the edge to reach the contour is along the direction perpendicular to the edge orientation. Thus the maximization process will find the accurate contour location more quickly by moving the edge to the search direction. The search strategy also helps to maintain uniform spacing between edges on the same contour and prevents them from being attracted to those with high edge strength and colliding into a single point.

Figure 3 gives a pseudo code for this procedure. We used Nitzberg-Shiota's gradient operator ([18]) to obtain an initial edge configuration. For each edge element, it requires 3 evaluations of S_i or equivalently F_i . This is a significant reduction from $N_o N_l$.

The edge configuration resulting from this maximization procedure is important for determining object contours in noisy images. Our final goal is to use the resulting edge configuration to obtain high-resolution edge contours.

4 Contour Extraction

It is very useful in many vision applications to extract contours at sub-pixel accuracy. For example, an effective sub-pixel contour extraction process can aid

data analysis of low-resolution data, improve visual quality of image expansion, and increase spatial accuracy of matching algorithms.

Using the contour fragments from the edge localization process, we create a boundary contour that is continuous at a high resolution. We want to do in such a way that the grouping result is compatible to our visual perception, and the process is computationally efficient. Many edge grouping techniques have been developed so far. We found it beneficial to develop another one that is tailored to the particular information available to us for both computational and performance reasons. Several steps are necessary to obtain the result.

First, localized edges are resampled on the new finer lattice. Assuming that we are interested in extracting contours at the resolution ρ times higher in both horizontal and vertical directions than the original data, the size of the resulting contour image is $\rho \times \rho$ larger than the original. An edge, $e(x, y)$, is placed at $(\text{round}(x\rho), \text{round}(y\rho))$ of the new lattice, where $\text{round}(x)$ returns the integer closest to x . When multiple edges reside on the same lattice, only the edge with the largest $|e|$ is kept.

Second, edges are grouped into a contour based on proximity and continuation. Since the localization and resampling processes effectively reduce both ambiguity of edge location and the number of spurious edges, a simple perceptual organization technique works well for this task. Also since edges are distributed very evenly along the contour after the localization process, the search can be restricted to within a small neighborhood. We found that a search distance as small as ρ is often enough for our purpose.

For each edge, our grouping procedure searches in its neighborhood for two edges based on some proximity and continuation criteria. For the first pair, we choose heuristically and empirically the following quantity to measure the continuation of two edges. Denoting e_i as the current edge for the grouping process and e_j as a neighbor edge with which the grouping criteria is being evaluated, the continuation measure, μ_{ij} , is

$$\mu_{ij} = \cos^2(\alpha_{ij} - \angle e_i) \cos^2(\alpha_{ij} - \angle e_j). \quad (16)$$

The definition of α_{ij} is the same as before. Thus, the continuation measure ranges between 0 and 1. It is 1 when the orientation of each edge is either the same with α_{ij} or different by π . It is 0 when one of the edge is perpendicular to α_{ij} . The measure varies smoothly between the extrema.

For the second pair, we also take the smoothness of a contour formed by the first pair and e_j into consideration. Then the continuation measure for the second pair, $\hat{\mu}_{ij}$, is the product of the smoothness measure and μ_{ij} .

$$\hat{\mu}_{ij} = \sin^2(0.5 * (\alpha_{i1} - \alpha_{ij})) \mu_{ij} \quad (17)$$

where α_{i1} is the slope of the line connecting the first pair. Again, the measure ranges between 0 and 1. It is 1 when three edges are colinear and point the direction of the line connecting them. It is 0 when either $\mu_{ij} = 0$ or $\alpha_{i1} = \alpha_{ij}$.

The proximity measure is incorporated into the order of the search. We first start the search in the neighborhood whose chessboard distance is 1 from the

current lattice site (i.e. $\max(|i_x - j_x|, |i_y - j_y|) = 1$). If the maximum of the continuation measure in this neighborhood is above some threshold ζ , then the site associated with the maximum is selected and the search stops. When no sites has continuation measure above ζ , the search continues in the neighborhood at distance 2 then 3 and so on until the distance reaches over pre-defined maximum, D . Advantages of this strategy are that the number of search to find a match is smaller than having the fixed search area and the formulation of a 'goodness' measure is simplified as only the continuation measure needs to be considered. The disadvantage is that the search can miss the 'best' match when a decent match is detected before.

Note that the grouping process is not symmetric, i.e. e_i selecting e_j as a grouping pair does not guarantee e_j selecting e_i . This asymmetric property is used to form T-junctions.

The next step is to interpolate a pair of edges to form a contour. Typically, the distance between a pair of grouped edges is small, and we found a simple first-order polynomial interpolation is visually acceptable for 4×4 expansion used in our experiments. For higher expansion rates, higher order polynomials may be required. One alternative is to use an *Essentially Non-Oscillatory* interpolation scheme for better preservation of corners and junctions [26]. Another possibility is to use the F_i field so that the curve traces the ridge of the field. These alternatives will give smoother interpolation but are more computationally demanding.

For every grouped pair of edges, an 8-digital straight segment is drawn and lattice sites on the segment including both starting and ending edge sites are marked. Then a contour is defined as a 8-connected component of marked sites, and the contour length is defined as the number of sites contributing to the contour. For details of digital straight segments and how to draw them, see [16].

Now F_i is computed at every contour point. When F_i is below some threshold η , the point is removed from the contour. After the thresholding, the procedure finally removes contours whose lengths are smaller than some threshold L .

5 Experimental Results

5.1 Edge Localization

At low resolution, our maximization procedure effectively refines edge locations while removing spurious edges at the same time. Figure 4 shows synthetic test images. One is without noise and the other with additive Gaussian noise. The signal to noise ratio of the noisy image is 1.5. The image size is 64×64 pixels. Throughout the experiments, the following set of parameters is used.

$$\beta = 0, \sigma = 0.5, \epsilon = 0.25.$$

Figure 5 shows initial configuration of each test image. Edges are thick mainly due to 3×3 mask used in the Nitzberg-Shiota operator. Thus there is 3-pixel ambiguity in edge location even for the clean image.

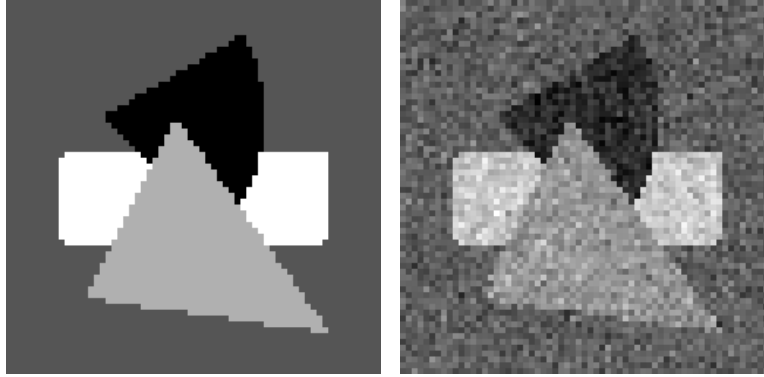


Fig. 4. *Synthetic Test Image.* The actual size of the images is 64x64 pixels. They are expanded by pixel duplication for viewing.

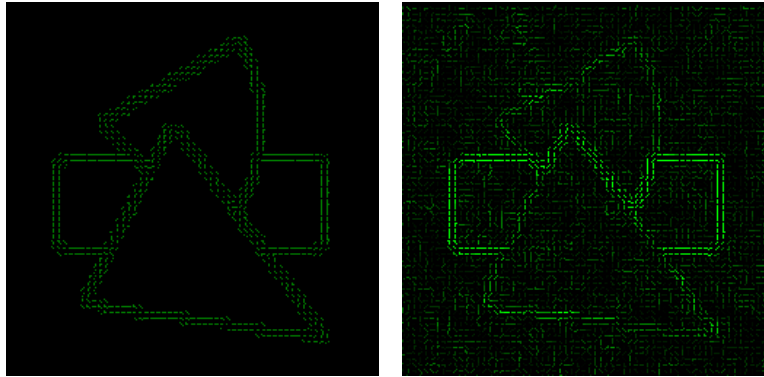


Fig. 5. *Initial Edge Configuration.* The figure shows the initial edge configuration of test images shown in Figure 4.

Figure 6 shows the results of the maximization procedure. It is evident from the result of the clean image that the procedure effectively resolved the ambiguity of edge location and provided more accurate locations of the edges. For the noisy image, the procedure combined random noise edges and produced some additional patterns. However, due to the random nature of these edges, the patterns are shorter in length than those formed by actual contour edges. They also tend to contain edges whose S_i is small because of high curvature at the locations. By removing edges with S_i smaller than some threshold value, random patterns are broken into even smaller pieces, and the true patterns and random patterns can be separated effectively based on the contour length.

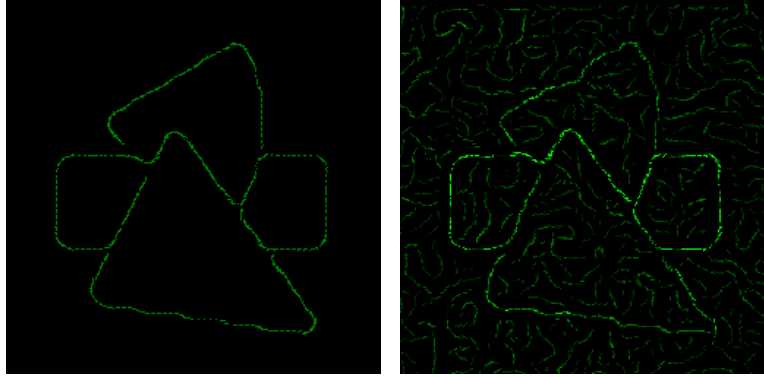


Fig. 6. *Localized Edge Configuration.* The figure shows the edge configuration of test images shown in Figure 4 after 10 iterations.

5.2 Contour Extraction

The results of the contour extraction process are shown in Figure 7. Parameter values are given in the figure caption. For the clean image, the complete contour boundaries are extracted with high localization accuracy. For the noisy image, noise edges are effectively removed while most of actual boundaries are extracted. For the clean image, a larger D is used to connect edges at junctions.

This contour extraction process is applied to natural images. Results are given in Figure 8. The process extracted subtle features without being affected by random noise. For example, with the house image, it delineated the outline of the roof more completely than other edge extraction techniques we tested. With the seagull image, it extracted the pattern of the feather without picking up relatively strong random patterns in the background.

5.3 Comparison

For comparison, Canny's edge detector ([2]) is applied to the synthetic images. The result is shown in Figure 9. The detector uses Gaussian smoothing followed by a gradient operator for detecting edges. Such a simple linear operator fails to distinguish true boundaries from random noise. As the amount of smoothing is increased, the number of spurious edges decreases and at the same time the real surface boundaries are removed as well.

Overall, the whole process of sub-pixel contour extraction on a 64x64 image expanded to 256x256 with 20 relaxation iterations took 25 seconds on a 300MHz SGI O2. Note that our code is not optimized for ease of maintenance (we implemented it in C++ using vector STL that contains large overheads in both speed and memory usage) and we believe that a significant amount of improvement on the speed can be achieved. The most computationally intensive part of the process is the relaxation, which consumed 90% of computation time.

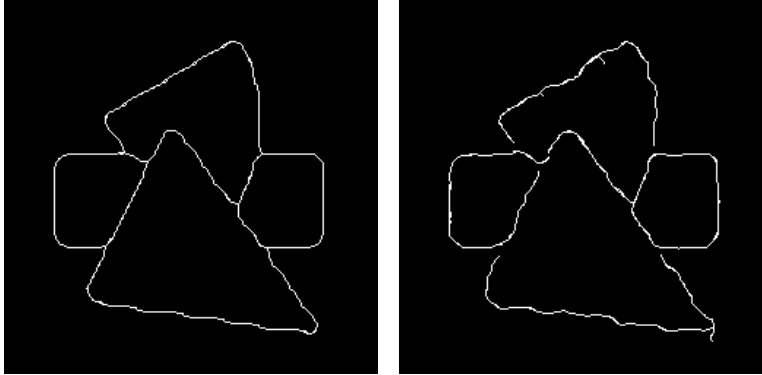


Fig. 7. *Result of Contour Detection.* Contour detection is applied on 64x64 synthetic images at the 256x256 resolution. The following set of parameters is used. Left: $\zeta = 0.5$, $D = 10$, $\eta = 1.0$, $L = 75$ Right: $\zeta = 0.5$, $D = 4$, $\eta = 1.0$, $L = 75$

6 Discussion

In this section, we give brief discussion on other contour extraction works and some neurological evidences relevant to our development. It also addresses a drawback associated with our technique and a future extension to fix it.

As shown by Equation (14), the maximization of r_{or} can be done based on a sum of vectors that are induced by edges. Similar vector field based edge extraction techniques have been proposed in the past. [3, 5, 6, 19] These techniques use the field representation to measure the *saliency* of features while ours actively reconfigures the edges by the probabilistic relaxation process. This active reconfiguration process tends to increase the saliency of structured patterns more than random ones, resulting in clearer separation of two types of patterns. We also allow the reconfiguration to take place in terms of the spatial location to further increase the saliency of structured patterns and at the same time provide their accurate locations.

The field based techniques can be associated with long-range interactions of biological neurons in the visual cortex. With this view, the compatibility function represents long-range interconnection patterns to facilitate contour completion with their neighbors. *Association field* by Field et al. [4] and the oscillatory intracortical network by Li [15] are some of examples to model the contour integration process of the human vision using long-range neural interconnections.

The research on detecting edges at sub-pixel accuracy under noisy condition dates back to Hueckel's work [8]. Typically, edge locations are estimated in the continuous domain based on theoretical modeling of image formation and edge detection processes. Noise is often handled either by explicit thresholding on the strength of edges or implicit smoothing by interpolation functions.

Our technique also employs image formation and edge models that are captured in the compatibility function. Instead of applying thresholding or smooth-

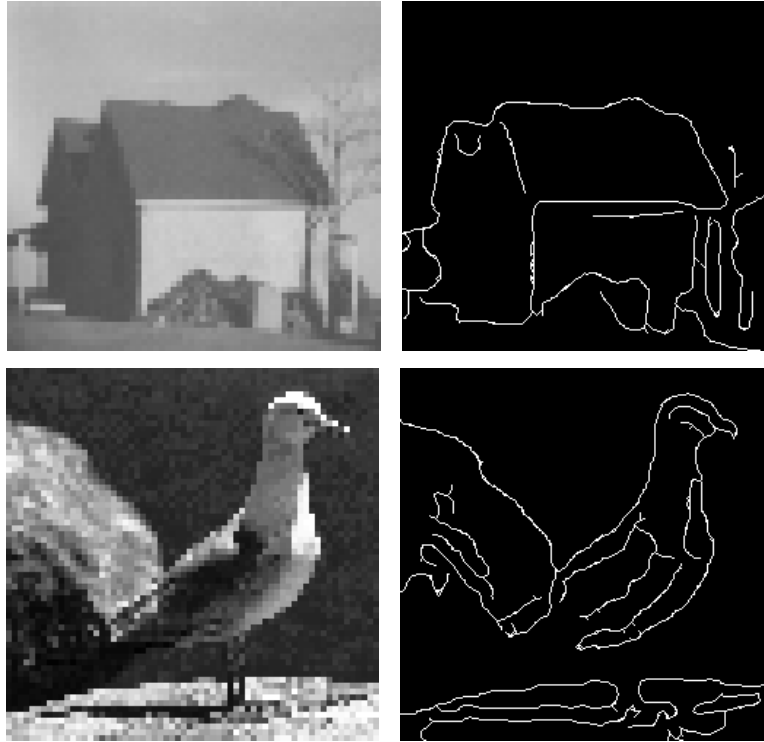


Fig. 8. *Result of Contour Detection on Natural Images.* Contours are detected at the resolution 4x4 times higher than the original. The left column shows the original images expanded by 4x4 using pixel duplication. The right column shows the corresponding results. Parameters used are $\zeta = 0.1$, $D = 4$, $\eta = 0.5$, $L = 30$ for the house and $\zeta = 0.1$, $D = 4$, $\eta = 0.5$, $L = 50$ for the seagul.

ing, it delays handling noise until edges are reconfigured based on the relaxation process. The process is effective in isolating noise edges from actual object boundaries as seen in Figure 6, and it becomes easier to separate them by simple heuristical rules as demonstrated in Figure 7.

A drawback of the unambiguous probabilistic relaxation for handling edges is that the technique tends to round off sharp corners and junctions due to its deterministic nature of the PDF specifications. A natural extension of this work is to allow more general specification of their PDFs so that corners and junctions can be represented with multi-modal PDFs. However, the extension loses the simplicity of the compatibility function design.

The system with the above extension resembles the hyper-column architecture proposed to model the V1 area of the human visual cortex [13, 23]. There has not been any development of systems at a practical level based on the architecture due to difficulties in designing reentrant and recurrent inter-neural



Fig. 9. *Result of Canny Edge Detection.* Canny edge detection is applied on synthetic images of 4 at the 256x256 resolution. The original 64x64 images are expanded using bi-linear interpolation prior to the edge detection operation.

connections. Our work bridges two extremes of pure neurological and engineering approaches and may give insight in the development of a neuro-morphological contour extraction system that is more faithful to the neurological structures.

7 Conclusion

The paper described a feature extraction process using unambiguous probabilistic relaxation to conduct local pattern analysis of an image. Through local iterative interactions controlled by the relaxation process, globally consistent patterns emerge at sub-pixel accuracy while noise is suppressed to form less consistent patterns. By post-processing the patterns based on the consistency measure derived from the support function, features can be separated from noise. Our formulation of the compatibility function allows efficient relaxation process requiring only 3 evaluations of the support function for each edge per iteration.

We developed a contour extraction procedure on a super-resolution lattice based on the relaxation result and a perceptual organization technique. The effectiveness of the procedure is demonstrated on both synthetic and natural images. Comparison with Canny operator shows superior performance of our technique in terms of noise robustness and localization accuracy.

References

1. L. Alvarez, P-L. Lions, and J. M. Morel. Image selective smoothing and edge detection by non-linear diffusion. ii. *SIAM Journal on Numerical Analysis*, 29:845–866, 1992.
2. J. F. Canny. A computational approach to edge-detection. *PAMI*, 8:679–700, 1986.
3. J. H. Elder and S. W. Zucker. Computing contour closure. In *ECCVPR'96*, pages 399–412.

4. D. J. Field, A. Hayes, and R. F. Hess. Contour integration by the human visual system: Evidence for a local "association field". *Vision Research*, 33:173–193, 1993.
5. S. Grossberg and E. Mingolla. Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentation. *Perception and Psychophysics*, 38(2):141–171, 1985.
6. G. Guy and G. Medioni. Inferring global perceptual contours from local features. *IJCV*, 20(1):113–133, 1996.
7. E. R. Hancock and J. Kittler. Edge-labeling using dictionary-based relaxation. *PAMI*, 12(2):165–181, 1990.
8. M. Hueckel. A local visual operator which recognizes edges and lines. *JACM*, 20(4):634–647, 1973.
9. R.A. Hummel and S.W. Zucker. On the foundations of relaxation labeling processes. *PAMI*, 5(3):267–287, 1983.
10. B. Jawerth, P. Lin, and E. Sinzinger. Lattice boltzmann models for anisotropic diffusion of images. *JMIV*, 11(3):231–237, 1999.
11. J.V. Kittler and J. Illingworth. Relaxation labelling algorithms – a review. *IVC*, 3:206–216, 1985.
12. I. Kovacs. Gestalten of today: Early processing of visual contours and surfaces. *Behav. Brain Research*, 82:1–11, 1996.
13. T. S. Lee. A Bayesian framework for understanding texture segmentation in the primary visual cortex. *Vision Res.*, 35(18):2643–2657, 1995.
14. S.Z. Li, H. Wang, K.L. Chan, and M. Petrou. Minimization of mrf energy with relaxation labeling. *JMIV*, 7(2):149–161, 1997.
15. Zhaoping Li. A neural model of contour integration in the primary visual cortex. *Neural Computation*, 10(4):903–940, 1998.
16. S. Marchand-Maillet and Y. M. Sharaiha. *Binary Digital Image Processing - Discrete Approach*. Academic Press, 2000.
17. J.L. Mohammed, R.A. Hummel, and S.W. Zucker. A gradient projection algorithm for relaxation methods. *PAMI*, 5(3):330–332, 1983.
18. M. Nitzberg and T. Shiota. Nonlinear image filtering with edge and corner enhancement. *PAMI*, 14(8):826–833, 1992.
19. P. Parent and S. W. Zucker. Trace inference, curvature consistency and curve detection. *PAMI*, 11(8):823–839, 1989.
20. P. Parent and S.W. Zucker. Radial projection: An efficient update rule for relaxation labeling. *PAMI*, 11(8):886–889, 1989.
21. M. Pelillo and M. Refice. Learning compatibility coefficients for relaxation labeling processes. *PAMI*, 16(9):933–945, 1994.
22. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990.
23. U. Polat and D. Sagi. The architecture of perceptual spatial interactions. *Vision Research*, 34(1):73–78, 1994.
24. A. Rangarajan. Self-annealing and self-annihilation: unifying deterministic annealing and relaxation labeling. *PR*, 33(4):635–649, 2000.
25. A. Rosenfeld, R.A. Hummel, and S.W. Zucker. Scene labeling by relaxation operations. *SMC*, 6(6):420–433, 1976.
26. K. Siddiqi, B.B. Kimia, and C.W. Shu. Geometric shock capturing eno schemes for subpixel interpolation, computation and curve evolution. *CVGIP: Graphic Models and Image Processing*, 59(5):278–301, 1997.
27. E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice-Hall, 1998.